

Session d'exercices – Structures et pointeurs

Echantillonnage

En traitement du signal, on sait qu'un signal échantillonné a une certaine fréquence d'échantillonnage peut être parfaitement reconstruit si la fréquence du signal est strictement inférieure à la moitié de la fréquence d'échantillonnage. On appelle cette fréquence maximale la fréquence de Nyquist. Dans cet exercice, nous définirons deux structures représentant les caractéristiques d'un signal sinusoïdal simple et d'un échantillonnage et utiliserons ensuite ces structures pour vérifier la théorie en échantillonnant des signaux de fréquences diverses. A noter que dans cet exercice, un signal complet est défini comme étant une somme de signaux sinusoïdaux.

Si la théorie sous-jacente n'est plus très claire pour vous, vous trouverez des informations sur les pages suivantes :

1. Signaux sinusoïdaux
 2. Echantillonnage
 3. Théorème d'échantillonnage
1. [Difficulté : *] Écrivez la structure **Sinusoide** qui doit contenir 3 champs représentant les 3 constantes inhérentes d'un signal sinusoïdal pur : la fréquence, l'amplitude et le déphasage.
 2. [Difficulté : *] Écrivez la structure **Echantillonnage** qui doit contenir 2 champs représentant l'instant auquel l'échantillonnage commence et sa fréquence.

Continuez en écrivant les fonctions suivantes :

3. [Difficulté : **] Écrivez la fonction


```
double sin_pure(double t, struct Sinusoide s),
```

 qui prend en argument un réel et une structure **Sinusoide** et retourne la valeur de la sinusoïde à un instant *t*. Si l'utilisation des 3 constantes d'une sinusoïde n'est pas claire, se référer aux pages de théorie plus haut.
4. [Difficulté : **] Écrivez la fonction


```
double signal(double t, int n, struct Sinusoide signal[n]),
```

 qui prend en argument un réel et un tableau de sinusoïdes de taille *n* et retourne l'évaluation du signal au temps *t*. Il s'agit donc de sommer les évaluations des sinusoïdes à travers le tableau passé en argument en utilisant la fonction précédente.
5. [Difficulté : **] Écrivez la fonction


```
void echantillonne(int n, struct Sinusoide s[n], struct Echantillonnage*
    echant, int nb_echant, double echantillons[nb_echant]),
```

 Cette fonction prend en argument un Signal, représenté par un tableau de Sinusoïdes, un Echantillonnage et un tableau de réels où seront stockés les échantillons du signal. Pour ceci, il faut donc évaluer le signal *nb_echant* fois en partant du temps *t0* donné par la structure **echant** et à une fréquence donnée par la même structure.

.....

6. [Difficulté : *] Écrivez la fonction

```
double sinc(double x),
```

Qui retourne simplement le sinus cardinal $\text{sinc}(x)$ d'un réel passé en argument. Pour rappel, $\text{sinc}(x) = \sin(\pi * x) / \pi * x$ et $\text{sinc}(0) = 1$.

7. [Difficulté : **] Écrivez la fonction

```
double reconstruction(double t, struct Echantillonnage echant, int nb_echant,
double echantillons[nb_echant]),
```

Qui, pour un Echantillonnage et un tableau d'échantillons (typiquement calculés au préalable par la fonction `echantillone`), retourne la valeur du signal reconstruit à l'instant `t`. Ceci est fait en utilisant la formule standard de reconstruction d'un signal, modifiée pour correspondre à cet exercice :

```
valeur[i] = echantillons[i] * sinc(echant.fe * (t-echant.t0) - i );
```

Il s'agit donc de sommer ces valeurs à travers le tableau d'échantillons passé en argument.

Une fois toutes ces fonctions écrites..